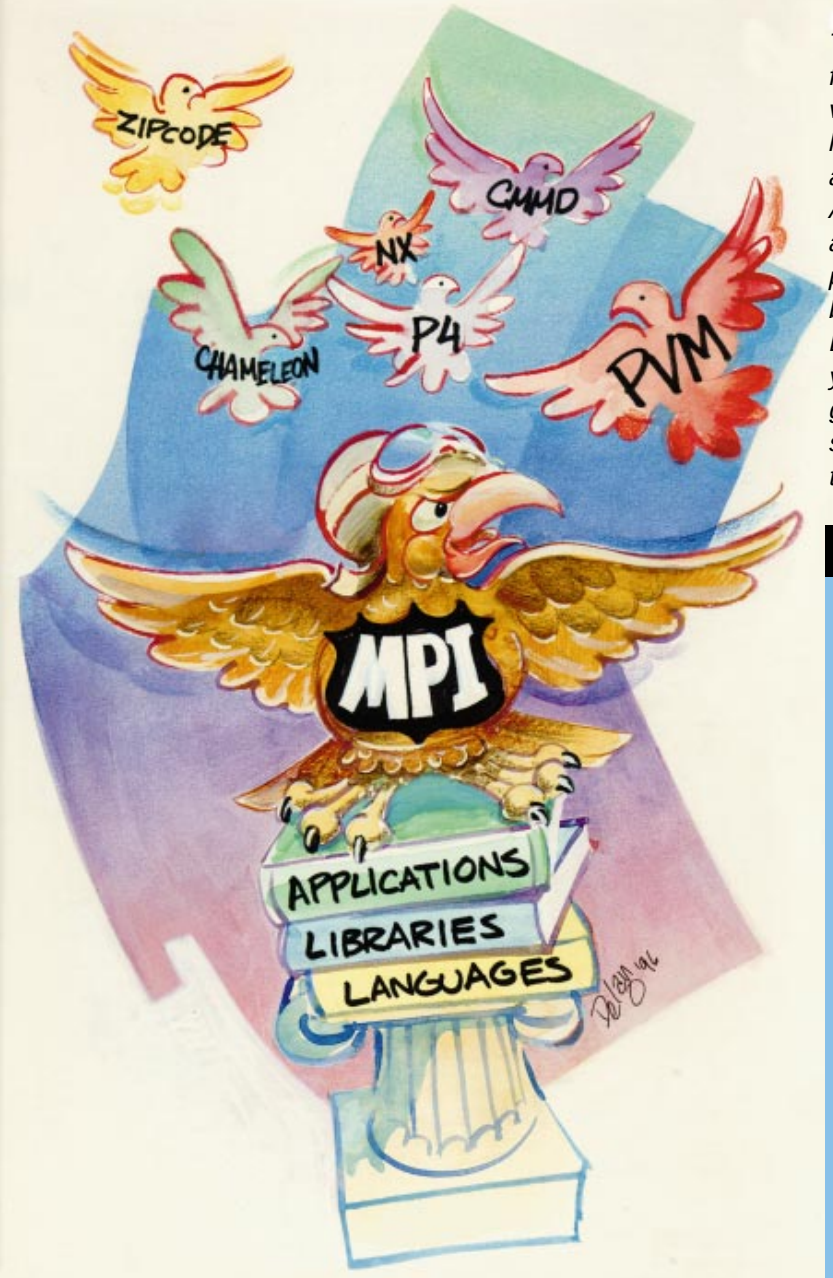
BITS
computing&communications news

JULY 1996

COMPUTING, INFORMATION, AND COMMUNICATIONS (CIC) DIVISION • LOS ALAMOS NATIONAL LABORATORY



The Message Passing Interface (MPI) Standard for parallel computing combines the work of various message passing research efforts and libraries as well as vendor software into one agreed upon collection of powerful functionality. Applications, libraries, and higher level languages are being built on top of MPI. Designed for high performance as well as portability, MPI can be both easy to use and confusing. The Parallel-Distributed Computing Team of CIC-8 can help you sort through the confusion and devise programming strategies that take advantage of the specific characteristics of the various implementations of MPI. See the article on page 5 for details.

Inside this issue

CIC Customer Service Center . . . . . (505) 665-4444 or cichelp@lanl.gov

Integrated Computing Network (ICN)

Consulting:

Centralized scientific and engineering computing .....consult@lanl.gov or 7-5746

Lab-wide administrative and business systems .....labwide@lanl.gov or 7-9444

Passwords (required for access to ICN) .....validate@lanl.gov or 5-1805

Central Computing Facility (CCF) 7-4584

Advanced Computing Laboratory (ACL) 5-4530

Desktop Support Center (DSC) 7-4357 (7-HELP)

For PC questions: PC-help@lanl.gov or 7-9372

For Macintosh questions: Mac-help@lanl.gov or 5-1361

For UNIX questions: UNIX-help@lanl.gov or 5-2220

For groups with CIC-2 support contracts: 5-2220

Telephone Services Center 7-3400

(includes voice mail)

Computer training

Lab-wide systems support training.....7-9444

Computer/workstation training.....7-9399

Personal computer training.....7-9071

Microcomputer support facility seminars.....7-4357

(Macintosh/IBM software, lending library)

List of Forms

Responsible Use of the Internet

The Laboratory's Internet is frequently thought of as the machines, software, and connections that serve information to the Laboratory and to the world at large. While true, this view is not complete.

In addition to the above, the Laboratory's Internet is also the resource that enables us to reach beyond the Laboratory to view remote information and collaborate with others. The Internet is not just what we see, but also the tool that enables us to see it.

The Laboratory's Internet community therefore extends beyond those of us who maintain World Wide Web sites to include everyone who uses Netscape or a similar tool to make use of the Web. The Internet is a resource shared by everyone within that community, and the way we use it has the potential of affecting everyone else within that community.

This makes the subject of responsible use of the Internet fundamentally different from responsible use of isolated machines. If somebody uses his/her desktop PC to view pornography from a CD, for example, then that person's activity is an isolated event that is not likely to affect the rest of us. If, however, that same person used the Web to view those same images, then the resource shared by the entire community would be threatened.

Responsible use of the Internet includes an ethical component, a responsibility to the others who rely on the shared resource.

Whenever we use Laboratory computers, we must remember that these resources do not belong to us. They are provided to us by the U.S. taxpayers, via the Congress, and they are provided with certain restrictions on how we may use them. Regardless of how much freedom we may wish to assert, as long as we are generating taxpayer-supported information on taxpayer-provided machines, there are boundaries we must live within.

The Internet, especially as expressed through the World Wide Web, offers us amazing professional and personal capabilities. The Internet has permitted me, for example, to collaborate with people from California to Washington, DC, in pursuit of professional goals, with the results being shared with others as far flung as Singapore and Poland. At home, using non-Laboratory resources, the Internet has also permitted me to establish "cyber-friends" ranging from Australia to Canada to Ireland, pursuing interests such as flags of the world, Celtic Knotwork, and peace in northern Ireland.

There is an important distinction, however, between professional work done with Laboratory resources and personal interests pursued with private resources. What any of us does at home with our own resources is our own business (assuming we are not revealing sensitive or classified information); what any of us does at the Laboratory, however, is bound by "official use" restraints. While a discussion of Celtic Knotwork might by some stretch of the imagination qualify as a "gray area," it does not clearly advance the Laboratory's goals; hence, it is not something to be done on Laboratory machines.

If we use the "information superhighway" as our metaphor, we can think of "incidental activities" as akin to slipping a little bit over the speed limit. Perhaps we share certain interests with a colleague—cooking, rock climbing, music—that are discussed within the broader context of our professional collaboration. As long as these discussions take place within the broader context, there are few people who are likely to object (although questions may arise if these discussions do not have the broader context—why, for example, should the American people be asked to pay for improving our cooking skills?).

There are also activities that are so clearly prohibited that they are akin to driving the wrong way down the freeway. We all know, for example, that several national laboratories have already endured significant scandals involving the storing or accessing of pornographic materials from government machines. A somewhat smaller group of us is aware of the controversy currently arising from certain U.S. Senators and Representatives using their government-funded Web sites for clearly partisan (i.e., prohibited) activities. Either of these examples places the rest of the respective communities at risk.

A Few Notes on Appropriate Use

• As a practical consideration and an acknowledgment of the interconnected nature of the Web, no one is likely to mind if you occasionally and accidentally stumble into a questionable site. Most of us, including myself, have followed links that seemed innocent only to find they lead to unacceptable material. Problems can arise, however, with repeated and ongoing misuse.

• There is a difference between reading "political" information and actively participating in political discussions. Simply reading an informational article can frequently be justified as incidental to our work duties, especially if it's an article about events that might potentially affect the Laboratory. Offering opinions or providing links to clearly partisan sites, however, crosses the line.

• Examples given in this article are for the purpose of illustration and should neither be considered examples of current Laboratory activity nor considered assurance that specific activities are acceptable. Common sense and good judgment remain key.

Technical Realities

The way the Web works is that any information you call up on your browser (e.g., Netscape, Mosaic) is transferred to your desktop machine before it is interpreted and displayed. If you look at something, your machine is storing it. If you use a Laboratory machine to look at something prohibited, then that Laboratory machine is storing it. (Sticklers can purge their cache if they come across material they don't want on their machine, but that's generally not necessary for occasional, accidental accesses.)

Although it may seem Orwellian to some, the fact of the Web is that we maintain activity logs that give us a great deal of information about what our users are doing. This certainly isn't the limit of what can be done, but standard logs track information such as machine address, files accessed, and the text of unencrypted postings. This information is not maintained out of nosiness or malice, but out of the needs to validate transactions, to monitor the effectiveness of our Web sites, and to ensure that our own machines are adequately secure. Again, this isn't the limit of what can be recorded; it's just the starting point. Furthermore, perhaps even more significantly, these logs are maintained not only by the Laboratory, but by almost everyone else who runs a Web site.

The first implication of these logs is that everything we do is basically public. If I use a Laboratory machine to access a remote site, then not only is a record of that access available to the Laboratory, but it's also available to the remote site. If, then, I am performing a prohibited activity such as viewing pornography, then that information is available to both the Laboratory and the provider of the pornography. In turn, if the provider of the pornography chooses to provide that information to a newspaper or magazine (e.g., the Wall Street Journal or Newsweek), then the entire world can quickly learn that Laboratory machines are being used for that activity, which could have an immediate impact for everyone else who relies on the Laboratory Web to accomplish their work.

By the way—

• your machine address identifies you as coming from the Laboratory, and "spoofing" that address to pretend to be someone else is easily defeated;

• encryption can mask the content of your messages, but it doesn't mask where you're coming from or what you're accessing; and

• no amount of tricks obscures the fact that what any of us does on our network potentially affects all other members of this community.

The second implication of the logs is that Web site maintainers need to use the information with care. I don't think any of us are in the mood to embarrass people or put their careers at risk, but we do have access to information that can lead to troubling situations. Generally, the ethical approach is to separate the machines from the activity being performed. If someone is using a Laboratory machine to access medical files that indicate a possible medical condition, for example, we treat that connection as sensitive, even though the accessing of the files, when separated from the individual doing it, might serve as important justification for the value of our Web.

At the same time, however, if we see that someone is using a Laboratory machine to perform political activities (regardless of how accidentally we discover that information), then no matter how much we support free speech as individuals, we are also required to report that activity. This is generally not done to interfere with people, but out of both the requirement to uphold Laboratory policy and the need to protect others' ability to continue to use our Web. Remember, if we can see that somebody is performing a prohibited activity, so can others; for us to act to control that activity is far more preferable than others acting in a more public manner.

Similarly, as soon as we put anything up on the Web, it is available for audiences to find. The same capability that allows somebody in a scientific division, for example, to read and publicly criticize my work also enables me to read his criticism and publish a direct link to it (perhaps within the context of other, more complimentary links).

In "Illusions," Richard Bach writes, "Live never to be ashamed if anything you do or say is published around the world, even if what is published is not true." If we accept this philosophy and act accordingly, the Web is a powerful, friendly tool. If, however, we overlook the fact that the Web also makes such publication technically possible (regardless of whether we like it), then we run the risk of forgetting that what might seem like a "harmless pastime" can in fact have serious consequences for us and the rest of our community.

Think of the Web as a penultimate small town. We all gain the chance to collaborate and share information and become familiar with others in our community—far more widely than geography would otherwise limit us to. But our curtains are never really drawn, and our behavior should reflect that awareness.

IA Guidance

The Information Architecture Project has adopted a series of guidelines dealing with responsible use, publishing, links, and related issues associated with the Internet/Web. For example, IA-5507: Unclassified Internet Responsible Use Guidelines enumerates four guidelines for use of the Internet/Web:

1. Promote the sharing of information.

2. Protect sensitive and classified information.

3. Use the Internet for official purposes.

4. Demonstrate professional, ethical, and courteous use.

A general discussion of each of these points is included in the guideline. The fourth point, for example, includes

"Whenever you publish or send any information outside the Laboratory, remember that you are representing the Laboratory to the outside world. Strive for professionalism. Add appropriate qualifications if the material is not supposed to represent the Laboratory as a whole. Acknowledge ownership when others own information. Notify people if they're about to get large or unusual documents. And avoid heated comments and political disputes while utilizing Laboratory resources."

IA-6301 through IA-6307: Guidelines for Publishing on the Laboratory Internet/WWW provides Web authors and publishers with guidance about appropriate information, information categories, classification review and publications release, and standard Laboratory copyright notices and disclaimers.

In this series, for example, IA-6302: Appropriate Information for Internet/WWW Publication states that

"In general, information is considered appropriate for Internet/WWW publication if it is

- work related;
- non trivial;
- not an infringement of copyright or a violation of licensing agreements, other parties' proprietary rights, or other restrictions;
- appropriately controlled."

Again, each of the points is described in further detail within the guideline.

These and other IA materials are available on the Web at http://www.lanl.gov/projects/ia/ or look under "What's New" from the Laboratory home page. For a summary overview of Web-related guidance, select the "Web" button from the masthead or the "World Wide Web" option from the Summaries near the bottom of the page.

Additional areas addressed by the IA Project include desktop software, data formats, network architecture, communications protocols, and more. Input on IA work is sought and welcomed from people throughout the Laboratory community.

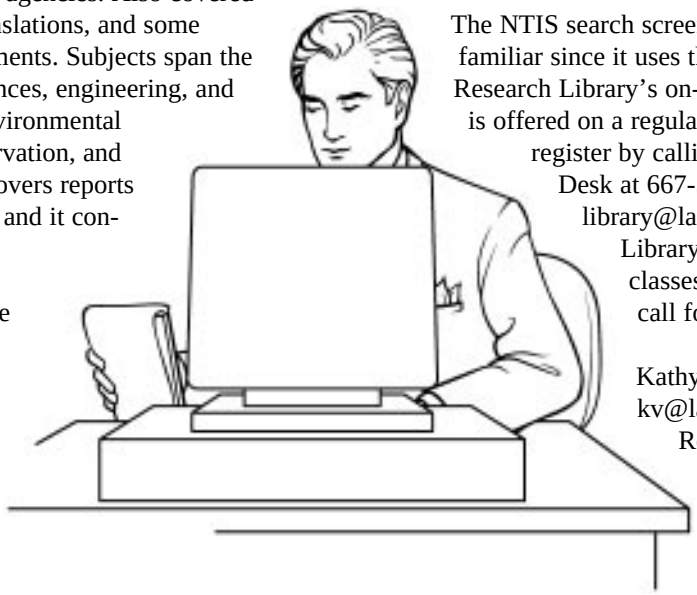
Tad Lane, tad@lanl.gov, (505) 667-0886
Information Architecture Standards Editor
Communications Arts and Services (CIC-1)

NTIS Research Database at Your Desktop

A database developed by the National Technical Information Service (NTIS) is now available, without charge, to the Laboratory community through the Research Library (CIC-14). The NTIS database contains listings and abstracts of reports published by the Department of Energy, Department of Defense, NASA, and many other agencies. Also covered are data files, software, patents, translations, and some reports prepared by foreign governments. Subjects span the physical, biological, and social sciences, engineering, and cross-disciplinary topics such as environmental pollution and control, energy conservation, and technology transfer. The database covers reports published from 1980 to the present, and it contains over one million records.

Many of the publications cited in the NTIS database are available locally within the LANL Research Library collections. When publications are not available locally, contact the Research Library for specific citations (667-5809 or library@lanl.gov).

To access the NTIS database, you can "telnet" to the Research Library's on-line catalog at library.lanl.gov, or find it on the Web at http://lib-www.lanl.gov/edata/ntis. Access is restricted to employees and contractors of Los Alamos National Laboratory.



The NTIS search screen interface should look familiar since it uses the same software as the Research Library's on-line catalog. Free training is offered on a regular basis, but you must pre-register by calling the Library Service Desk at 667-5809 or sending e-mail to library@lanl.gov. The Research Library also offers special group classes and orientations; please call for details.

Kathy Varjabedian,
kv@lanl.gov, (505) 667-3063
Research Library (CIC-14)

Orders for ISDN Suspended

According to Dave Hanger (800-872-4658; ex: 8903) of US West Communications, all orders for Custom or National ISDN (Integrated Services Digital Network) have been suspended pending resolution of US West's support of New Mexico's ISDN Tariff ruling: final order 95-769-TC dated 13 May 96. US West is protesting the flat rate, residential ISDN tariff of $40.86 per month and the implementation order. Basic residential telephone service is currently priced at $17.89 per month. Dave Hanger (dhanger@uswest.com) is with the Phoenix-based Home Office Data Team, which handles residential ISDN applications for US West's 14-state territory.

Interested parties may subscribe to the e-mail list of the New Mexico ISDN User's Group (NMIUG) by sending e-mail to ListManager@info-server.Lanl.GOV. The body of the e-mail should include either "subscribe nmiug" or "unsubscribe nmiug." Karl Pommer (CIC-4) is the acting spokesperson for NMIUG. For additional information, see the following URLs:

http://www.uswest.com/isdn/index.html
http://www.uswest.com/isdn/ut_single.html

Karl Pommer, kxp@lanl.gov, (505) 665-1641
Telecommunications Group (CIC-4)

Getting the Most out of MPI

Here's a basic shell for writing an MPI-based parallel program:

```
MPI_Init( &argc, &argv );

MPI_Comm_rank( MPI_COMM_WORLD,
               &my_process_id );

MPI_Comm_size( MPI_COMM_WORLD,
               &number_of_processes );

/* Application code, using the following
   to exchange data and perform global
   operations:

MPI_Send(...)
MPI_Receive(...);
MPI_Reduction(...);
MPI_Bcast(...);          */

MPI_Finalize();

/* Perhaps more user code independent
   of MPI */

exit(0);
```

However, you may want to consider using some of the 120 other functions included in MPI to (perhaps significantly) improve the performance and flexibility of your program. In this article we give an overview of some of the issues involved in using these additional functions, and since MPI is an interface definition, we discuss some issues regarding its various implementations. We also discuss some current proposals for MPI-2.

Message passing based programs are intended to do one thing—exchange data in a distributed memory parallel computing environment. (Message passing programs do, however, also run in shared memory environments.) Yet this concept can mean different things to different people due to application and computing environment specifics. Therefore, MPI provides a variety of ways to exchange data. Upon first glance these options can be confusing and possibly overwhelming. Therefore, we will attempt to sort them out here.

Additionally, it is important to keep in mind that MPI is an interface standard as agreed upon by a wide variety of vendors and researchers known as the MPI Forum. (The "MPI Standard" is available at the Web site listed at the end of this article.) However, the MPI standard cannot dictate exactly how tasks are accomplished. This allows the implementers to not only take advantage of the target architecture, but to also make decisions which may bias the implementation toward particular messaging schemes. (For example, many short messages versus a few long messages, as well as their definition of "short.") We will try to sort out these concepts. At the end of this article, we list a Web site that attempts to quantify the behavior of the various implementations of MPI. And since this Web site is dedicated to helping users, we also welcome contributions to this site as information becomes available.

Point-to-Point Communication: send/receive

MPI provides eight functions for sending a message from one process to another and two functions for receiving a message (plus ten functions for probing, testing, and waiting for messages). Deciding which functions to use requires knowledge of some terminology:

• Standard mode. Functions that are required to follow the "standard" MPI mode guarantee the safety of the operation. For example, when MPI_Send returns, the user is free to handle the memory location of the data being transferred. The standard mode receive, MPI_Recv, will not return until all the data it is receiving is safely in the memory space of that process; consequently, the user may begin using the data expected to be in the receive buffer. However, standard mode says nothing about how this is accomplished. A standard send may involve data buffering or not, may require that a matching receive be posted or not, or may require that the matching receive complete before returning or not. Additionally, these scenarios may unfold differently throughout the same execution of your program based on the availability of resources at the time the send is initiated. Resource availability is a function of the architecture and of how the implementor uses those resources. All of the above factors can significantly affect the performance of the operation.

• Blocking. The type of blocking used by a function determines when the function will return control to the user and the status of the data involved upon return. For example, a blocking send will not return until the data being sent is safely out of the user space. That is, the user is free to overwrite the address space of the transferred data. However, this requirement may be satisfied in different ways depending on how the data is physically transferred and protected. For example, a "locally blocking send" buffers the data before sending it, allowing for (possibly) a quicker return. Conversely, a "globally blocking send" will not return until a matching receive has been posted by the target process and the data is safely on the way to that target. The decision by an implementor as to

how to perform the block should be determined by the communication environment, specifically by the buffering requirements. Intermediate buffering of the message allows for local blocking, but it also incurs a performance cost. The actual cost of a globally blocking send is determined by the behavior of your application.

A nonblocking function will return before the operation is complete and implies nothing regarding the status of the data involved. When using nonblocking sends, MPI provides functions for (safely) checking the status of the data. For example,

```
MPI_Isend(...); /* send initiated */

— perform work independent of the memory
  address of the data being sent

MPI_Wait(...); /* Block until MPI_Isend is
                  complete, i.e. the user is
                  free to overwrite the memory
                  address space of the data.
                */
```

Results presented at the recent MPI Developers Conference show that this approach is working well for many implementations (specifically MPICH-based versions). And of course your results will also depend upon the amount of work you can do before calling MPI_Wait. So the decision on which method to use is a function of the implementation of MPI you expect to use and the setup of your program. (We plan to post experimental results on the Web page listed below.)

• Synchronous. Synchronous message passing requires that a matching receive be posted for the synchronous send. The send may be initiated before the receive is posted, but it will not return until the receive is posted. And again, the send may require that the receive complete before the send returns. The decision to enforce synchronicity is a function of the requirements of your application. Overall program performance is usually better with an asynchronous model, but many applications require synchronization, such as visualization programs.

• Ready. Ready mode communication requires the posting of a matching receive before the send can begin. This method is commonly used when "handshaking" between processes is needed (i.e., a process notifies another via send/recv that it is ready to receive data), and it should provide a performance advantage over an explicit handshake.

Various combinations of the above properties are provided by MPI. For example, MPI_Issend is a nonblocking synchronous send. The send is immediately initiated but will not return until a matching receive is posted. Yet the return of the send does not ensure that the data has been safely transferred. (MPI_Wait or MPI_Test are usually necessary.)

How Are Unexpected Messages Dealt With?

In asynchronous message passing, it is usually desirable to have messages sent before the receiving process has posted a matching receive. (Otherwise the process posting the receive is standing idle.) The question is "Where is the data located during this time?" The answer to this may determine the maximum size problem you can execute before performance deteriorates or your program hangs. For example, an implementation may store all "long" messages at the sending process until the matching receive is posted. Other implementations will store some of the messages at the receiver until some preset limit is reached. Some implementations let the user determine the definition of "long." All of these factors may influence your decisions regarding the use of nonblocking functions.

Message Polling

MPI provides several functions for checking message status. For example, when using a wild card receive (MPI_ANY_SOURCE or MPI_ANY_TAG) you may want to find out information concerning a received message before actually bringing it into your program space via an explicit call to a receive function. MPI provides for this capability by using various probe functions. (For example, MPI_Probe, MPI_Test, MPI_Wait, and relevant nonblocking variations.)

Some Properties of Point-to-Point Communication

Implementers are required to enforce certain characteristics with regards to sending and receiving messages. We discuss the important ones here. It is up to programmers to be aware of these characteristics and to design their programs accordingly. Descriptions of these characteristics follow:

• Message ordering within a specific communicator is required of MPI implementations. That is, if process 1 sends messages A and B (in that order) to process 2, process 2 will receive them in the order sent. Exceptions to this rule may occur when using wild card receives or multi-threading.

• Fairness is not guaranteed (even within the same communicator) when using source process wild card receives (source process = MPI_ANY_SOURCE). Suppose process 1 sends a message to process 2. Process 2 posts a matching receive.

However, this receive also matches an incoming message from process 3. MPI makes no requirement regarding which message will be received by process 2. Further, suppose that while process 2 is receiving the message from process 3, process 4 sends a message that has the same identifying characteristics as the message from process 1. Process 2 again posts the receive which now matches the message from both process 1 and 4. Again, MPI makes no requirement regarding which message will be received. Taken to the extreme, the message from process 1 may never be received by process 2, even though repeated matching receives have been posted.

• Progress. Suppose two processes attempt to exchange data as follows:

```
if ( my_pe == 0 ) {
   MPI_Send(...);
   MPI_Send(...);
}
else {
   MPI_Recv(...);
   MPI_Recv(...);
}
```

MPI requires that at least one of these two operations will be completed, independent of other actions in the system. That is, progress must be guaranteed.

However, if two processes attempt to exchange data in the manner shown below, progress is not guaranteed.

```
MPI_Send(...);
MPI_Recv(...);
```

Depending upon the implementation of MPI being used and the availability of resources at the time this sequence is executed, your program may hang. Resource limitations obviously affect buffering schemes. However, implementers are not instructed on how to manage resources, so an inferior implementation will run out of resources and quit before a quality implementation, which better manages these resources.

Note that the above examples use standard mode functions. However, the situations are not necessarily dependent upon those specific functions.

Collective communication

MPI provides a wealth of functionality with regards to collective communication (broadcasts, gather/scatter, reductions, and barriers). Here we discuss two important implementation issues.

• Determinism. For debugging purposes, as well as a requirement for certain applications, it is imperative that the reduction function return the same result given the same input. However, the MPI standard only "strongly recommends" that implementations enforce this practice. (Determinism requirements may prevent maximum optimization of performance.)

• Performance. An interface standard cannot define optimization requirements. A prime example of this involves collective communications, such as reductions. A quality implementation would utilize an optimized communication pattern such as a tree or some other pattern that takes advantage of the specific communication environment. Such an implementation would provide better scaling properties over an implementation that ignored this optimization.

Notes on Collective Communication:

• It is a mistake to rely on collective communications for synchronization. The only exception to this is when MPI_Barrier is implemented in hardware. The standard warns us that although synchronization may occur due to a certain implementation of MPI, it is a side effect that should not be counted on.

• The MPI standard requires that collective operations will not interfere with point-to-point communication. For example, a wild card receive will not inadvertently pick up a message involved in the collective operation.

Miscellaneous

• SMP Implementations. Message passing programs can execute in shared memory environments. However, we expect a good implementation of MPI to accomplish the data exchange using memory-to-memory copies rather than actual transmission of the data between processors. The initial implementations of PVM in this environment were not able to take advantage of this performance optimization due to the operating systems. However, this situation has been remedied, so we expect MPI implementations to also take advantage of the shared memory environment. An exception to this expected performance is one-sided communication (discussed below). There are hardware features as well as (proposed) MPI requirements that will require extra work in software which will be necessary to get this functionality to work at the expense of efficiency.

• Threads. Currently, there are no thread safe implementations of MPI. However, many implementers are actively researching this area and hope to provide a thread safe environment.

In fact, MPI provides functionality that could be quite valuable in thread based programming (for example, the MPI_Wait and MPI_Test family of functions). But programmers will still need to be actively aware of how these functions are designed to behave. (Thread issues are discussed throughout the MPI standard, usually in the "Advice to Users/Implementers" sections.)

• PVM-like Programming. Programmers used to the PVM style of programming are accommodated with MPI as shown in the example below:

```
pvm_initsend(...);
pvm_pack(...);
pvm_send(...);
```

becomes:

```
MPI_Buffer_attach(...);  /* Allocate buffer
                            space */
MPI_Pack(...);
MPI_Bsend(...); /* Or MPI_Ibsend */
```

Some reasons for using this style include quick conversion of PVM-based code to MPI, handling of noncontiguous data, and avoiding the strong typing requirement of MPI.

On the positive side, MPI_Buffer_attach lets the user allocate space for the entire buffer needed, avoiding that overhead in the pack function. (Repeated calls to PVM packing functions may result in repeated calls to malloc, perhaps adding significant overhead.) However, because of (possibly significant) performance degradation of this extra buffering step, programmers are encouraged to convert to nonbuffered MPI functions.

Distributed Resource Management

Even with 128 functions and more on the way with MPI-2, MPI does not provide all the functionality desired by all users.

Most prominently—

• Heterogeneity could utilize the optimized implementations of MPI on different platforms. The network versions allow for heterogeneity, but their performance on MPPs is, of course, less than that of the native versions.

• Resource management (fault tolerance, task migration, etc.). The MPI Forum determined that it could not provide an interface for resource management acceptable to all vendors. We are examining various options for providing these capabilities. Vendors are developing all-in-one packages (LSF, GlobalWorks). Additionally, an ORNL research group is working on a system that will manage the heterogeneous environment as well as coordinate individual tools (LSF, TotalView, etc.). Look for announcements regarding this issue in upcoming issues of BITS.

• Portable Ttiming Function. MPI-1 provides a portable timing function that returns wall clock time: MPI_Wtime().

MPI-2

Now that several MPI implementations have been in the user domain for almost two years, suggestions have been made regarding additions to the standard. Therefore the MPI Forum has reconvened to discuss these suggestions. The result will be MPI-2, which is expected to be completed by November 1996. Implementations should begin to appear shortly thereafter. Functionality added to the current version of MPI (MPI-1) includes the following:

• Spawning processes. MPI-1 simply allows for a static process startup, and the standard did not tell implementers how to accomplish this. This flexibility gave implementers advantages with regards to various optimizations, control, and process management. User feedback, however, showed a definite need for process control. Functionality for dynamic process startup and management will be contained in the MPI-2 specification.

• One-sided communication. One-sided communication provides the means for a process to "put" data into another target process without the participation of that target process. Likewise, a process may "get" data from another process without the participation of the source process. On the Cray T3D, for example, the infrastructure for accomplishing one-sided communication is provided (global shared address space). Implementers have been charged with the task of providing this functionality even if the computing environment does not provide the explicit infrastructure, and implementers have been provided the interface for doing so. Test implementations are now in use even before the final standard is determined, so this functionality should be available shortly after the formal proposal is accepted.

A couple of warnings: First, with regards to SMPs, a current hardware limitation will prevent the efficient implementation of one-sided communication within the SMP. And second, the efficiency of one-sided communication between loosely coupled computers is also expected to be poor.

• Other New Functions. F90 and C++ bindings, real time functions, nonblocking collective communication, external interfaces, and a variety of convenience functions are also in the proposal stages for MPI-2.

Currently, behavior of a program is undefined if messages are sent by a C process and received by a Fortran process, or vice versa. However, some implementations do define Fortran and C interaction (two known: network MPICH and SGI). The MPI standard states that "It is fully expected that many implementations will have such features, and that such features are a sign of a quality implementation." A proposal in MPI-2 would, if accepted, define this interaction.

Conclusions

We have discussed only a few of the 128 functions defined by MPI. However, we've probably covered enough for most users to write the basic functionality of their applications. Once this is done, it could be advantageous to thumb through the MPI standard looking for additional functions that could be utilized by your application. Although often quite technical, the standard often provides "Advice to User" sections which are valuable. Additionally, a few books are available for aiding the MPI programmer. For a list of books regarding MPI, see

http://www.epm.ornl.gov/~walker/mpi/books.html

This article also demonstrates that while MPI can simplify the basics of message passing programming, there are some "gotchas" that require careful attention to the details of how your application behaves. The safest MPI program simply uses MPI_Send and MPI_Recv for data exchange, yet this is usually not the way to get good performance. Our team is available for consulting regarding the use of MPI. For more information regarding MPI, including an evolving listing of implementation specific features, see our team's Web page at

http://www.cic8.lanl.gov/dist_comp2/MSGPASS/mpi.html

or contact us at ptools_team@lanl.gov.

Richard Barrett, rbarrett@lanl.gov, (505) 667-6845
Distributed Computing Group (CIC-8)

New CIC Annual Report Available On-Line

The 1995-1996 annual report for the Computing, Information, and Communications (CIC) Division at Los Alamos National Laboratory is now available on-line in portable document format (PDF) at

http://www.lanl.gov/Internal/divisions/cic/publications.html

This report includes summaries of all the groups and major projects within the division. A limited number of printed copies is available (send your request to jjmortensen@lanl.gov).

Special Edition of BITS Available On-Line

The special edition of BITS, Introduction to Computing at Los Alamos, is now available in HTML and PDF formats. This special edition is designed to assist both the uninitiated LANL computer user and those of us who want to enhance our existing knowledge.

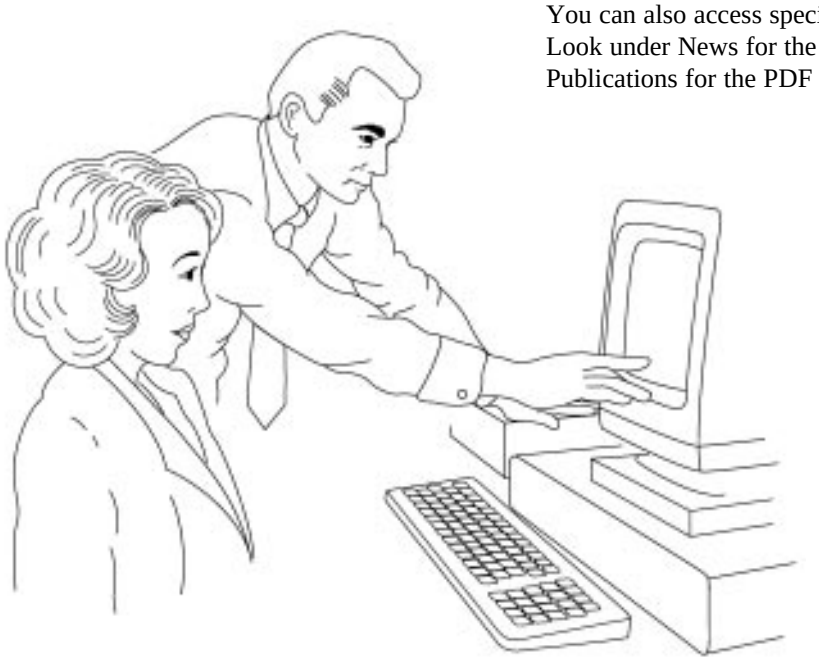
The URL for the HTML version is

http://www.lanl.gov/Internal/divisions/cic/bits/archive/96summer/specbits_contents.html

The URL for the PDF version is

http://lib-www.lanl.gov/la-pubs/00285952.pdf

You can also access special BITS from the CIC home page. Look under News for the HTML version and under Publications for the PDF version.



A Closer Look at MIME

In the beginning there were seven bits. That translated to 128 characters available for text—plain ASCII text. The world of computer communications was a serial world. People communicated to mainframes from "dumb" terminals using TELNET over a direct wire connection. Computer messages were short, text-based, and simple.

The original SMTP (Simple Mail Transfer Protocol) standard for Internet e-mail was set in 1982 and was based on this environment. These specifications only allowed for ASCII characters, the message could contain no lines longer than 1000 characters, and the message could not exceed a certain length. This standard was soon outgrown by the need for more involved messaging.

MIME Gets a Voice

In 1992, a revised standard for Internet mail was approved. This standard is called MIME—Multipurpose Internet Mail Extensions. The MINE standard is maintained by the IETF (Internet Engineering Task Force) and is based on Request for Comments (RFCs). The Internet standards document that specifies MIME is RFC 1521, which is built on and compatible with older standards, especially RFC 822.

New features provided by MIME include the following:

• Imposes no restrictions on line or message length,

• Accommodates character sets other than 7-bit ASCII (which allows for non-English language messages),

• Accepts binary or application specific files,

• Allows multiple "objects" to be embedded into a single message, and

• Provides support for other types of files, such as audio, graphic images, and video.

To quote from part one of the comp.mail.mime FAQ (Frequently Asked Questions) news group listing, "One of the best things about MIME is that it's a 'four-wheel drive protocol.' ... MIME was carefully designed to survive many of the most bizarre variations of SMTP, UUCP, and other Procrustean mail transport protocols that like to slice, dice, and stretch the headers and bodies of e-mail messages."

Trying to understand MIME from reading the 75-page RFC 1521 is a daunting task. Since MIME builds on older standards, this RFC deals a great deal with these older standards, field specifications, etc., and thus is not a straight forward presentation of the information. My thanks to Mark Grand for providing a concise and easy-to-read MIME overview, which helped greatly in writing this article. It is available in postscript and text formats at the Web locations shown below:

ftp://ftp.netcom.com/pub/md/mdg/mime.ps

ftp://ftp.netcom.com/pub/md/mdg/mime.txt

What Is MIME, Anyway?

Well, simply put, MIME is a way of specifying and encapsulating different set types of data, be it audio, video, a Microsoft Word file, or other binary files, and then placing them inside the old, text-only Internet e-mail standards in such a way that still conforms to the old standards.

Since most everything on your computer is in 8-bit (or binary) format, MIME needs to encode 8-bit data into a 7-bit format. Usually the encoding method known as Base64 is used. Base64 represents any sequence of three 8-bit bytes as four printable, 7-bit characters. (For more information about Base64 and other encoding methods, see "How to get Unattached to E-mail Attachments, Part 1: A Look at the Mac" in the May 1996 issue of BITS.)

With MIME, you can have any number of message "parts" encoded into a mail message. Each part stars with a "Content-Type:" header that specially tells the e-mail program what type of data is in that part of the message. One example would be "Content-Type: text/plain" for standard, old boring ASCII text. Another example would be "Content-Type: audio/x-wav" for WAV audio data. Each Content-Type actually has a type specification (e.g., Image), a slash, and a sub-type specification (e.g., gif). The Content-Type name itself, it's types and subtypes, and any parameter names defined in the MIME standard are case-insensitive.

These different MIME data types are registered with an entity called the Internet Assigned Numbers Authority (IANA). It is also possible to use non-registered MIME types. The type names for these begin with "x-" as in the "x-wav" example above. A listing of registered and unregistered MIME types can be found in part three of the comp.mail.mime FAQ. (You could make up your own data type specification and send it across the Internet so long as the e-mail package you were sending the attachment to also knew about that data type and knew what to do with it.)

The eight general Content-Type specifications are listed below. The name of the specification appears after "Content-Type:" and before the slash and sub-type (e.g., Content-Type: text/plain).

(1) Text—used with different character sets (like a foreign language) or with specialized formatted text like HTML (hypertext mark-up language) or RTF (Rich Text Format).

(2) Application—used to send specific types of application data or binary data.

(3) Message—used for encapsulating mail messages.

(4) Image—used for transmitting still image or picture data.

(5) Audio—used for transmitting audio, music, or voice data.

(6) Video—used for transmitting video data.

(7) Multipart—has many uses. It can combine several different types of body data parts into a single message. With Multipart/Parallel, for example, you could combine voice and video together.

(8) X-TypeName—used for experimental and future types or data.

If a message does not have a Content-Type field in its header, then the Content-Type is defaulted to "Content-Type: Text/plain; Charset=US-ASCII." When an e-mail reader finds a body part in a message with a Content-Type value it does not understand, it will generally treat it as equivalent to application/octet-stream; that is, a stream of binary data.

The Structure of a MIME Message

Every MIME encoding must begin with "MIME-Version: 1.0" on a line by itself (without the quotes).

Each individual part of a MIME message must start with a Content-Type header field, which specifies the type of data in that body part by giving the type and subtype identifiers. This heading can also give parameters that may be needed for certain data types. For example:

```
Content-Type: application/applefile;
        name="File Name"; type="MSWord5.1a"
```

The data type of Multipart has to have some way of keeping track of all its parts; that is, where one ends and the others start. For this reason, each body part is preceded by a special string called an encapsulation boundary, and the last body part is followed by a closing boundary. The encapsulation boundary is specified in a boundary="some text" line, where "some text" is a unique string. Some e-mailers pick a random sequence of numbers and letters for this string. When the encapsulation boundary is used to start a body it simply lists the specified text string (without the quotes) preceded by two hyphens. The closing boundary is simply the specified text string (without the quotes) preceded and followed by two hyphens. (The example below will make this clearer.) Things can get even more confusing when a Multipart part has another Multipart part as one of its parts. In this case, there would be two different text strings used as the two encapsulation boundaries.

Below is an MIME example adapted from the "MIME Overview" document by Mark Grand. It is an example of the Multipart/Alternative type, which is syntactically identical to the Multipart/Mixed type that allows for different body parts to be strung together. But here, each part is an "alternative" version of the same information. A mail reader so equipped would recognize that the content of the parts is interchangeable and should either choose the "best" type based on the user's environment and preferences, or offer the user the available alternatives. Generally, choosing the best type means displaying only the last part that can be displayed. This may be used, for example, to send mail in a fancy text format in such a way that it can easily be displayed anywhere.

```
From:  Irma User <iruser@lanl.gov>
To: Rick User<uruser@somewhere.com>
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary=boundary42

--boundary42
Content-Type: text/plain; charset=us-ascii

... plain text version of message goes here ...

--boundary42
Content-Type: text/richtext

... richtext version of same message goes here...

--boundary42
Content-Type: text/x-whatever

... fanciest formatted version of same message goes here

--boundary42--
```

In the above example, users whose mail system understood text/x-, whatever format, would see only the fancy version, while other users would see only the richtext or plain text version, depending on the capabilities of their system.

If you were sending a binary file that had to be Base64 encoded, it would have a "Content-Transfer-Encoding: Base64" line inserted after the "Content-Type:" line, as in the example below.

```
MIME-Version: 1.0

Content-Type: Multipart/Mixed;
        Boundary="-- next item ---"

--- next item ---
Content-Type: Text/Plain
```

This is the text part of the message. Here you would tell the person to look at the enclosed attachment, etc.

```
--- next item ---
Content-Type: Application/X-Lotus-Notes;
        Name="Important File"
Content-Transfer-Encoding: Base64

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAA
AOwADAP7/CQAGAAAAAAAAAAAAAAABAA
AQAAAAAAAAAAEAAAAgAAAAEAAAD+////A
A AAAAAAAAD
//////////////////
(etc., etc.)

--- next item ---
```

In Summary

MIME is an involved, useful, and extensible standard. It was built on top of outdated standards and managed to adhere to those standards without being limited by them. The parts of a MIME message you receive should decode without a fuss so that the behind-the-scenes structure of MIME (as discussed in this article) should not be apparent.

You do, however, see the structure of a MIME message if a MIME-encoded message gets sent to a non-MIME e-mail package. You would then need to manually decode these attachments as discussed in the two previous BITS articles "How to get Unattached to E-mail Attachments" Parts 1 and 2. For Mac and PC users, Eudora, Lotus Notes, and Netscape 2.0 are MIME compliant. For UNIX users, Netscape 2.0 and Pine (and others) are MIME compliant. (Look at http://w3.lanl.gov/projects/ia-lanl/areas/email/e-results.html for a listing of the capabilities of many e-mail programs.)

It's worth mentioning that the encoding method of Mac BinHex is a MIME standard. An example of its use would be

```
Content-Type: application/mac-binhex40;
        name="file name.hqx"
```

where file name.hqx is the name of the encoded file. A problem with BinHex is that some e-mail gateways can't decode it. (This has been the case with the Lab's Lotus Notes gateway, for instance, but this is currently in the process of being upgraded.) A good article on MIME and BinHex is located at the following URL:

ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/application/mac-binhex40

With MIME's capabilities for expansion, it's only a little farfetched to imagine that in the future the Content-Type of Application/Transporter would contain some rather unusual and important data. We can just hope that the problem of corrupted data would be solved by then.

For questions about MIME and e-mail attachments, you can call the CIC-6 ICN Consultants at 5-4444 or call 7-HELP (7-4357).

John Layne, jpl@lanl.gov, (505) 665-5090
Desktop Group (CIC-2)

Vendor Computer Training

The Customer Service Group (CIC-6) supports vendor training in technical computing areas such as programming languages, system administration, networking, and World Wide Web development tools. The support provided by CIC-6 can be as limited as providing the appropriate facilities for a specific group or as extensive as coordinating training functions such as system administration, vendor acquisition, EDS administration, and class facilitation. The table below lists classes that are either currently being offered or are available on request. An expanded list of classes that are potentially available can be viewed on the Internet at

http://www.lanl.gov:8010/computer-information/ComputerTraining/Vendor.html

To request registration in any vendor course or for general assistance with vendor training, please contact the CIC-Division Vendor Training Coordinator at (505) 667-9399 or send e-mail to cic6-train@lanl.gov.

*Cost per student will vary depending on the total number of students enrolled in the class.

Course Title	Date	Time	Cost	Course Number
C Programming (Beginning)	**8/5-9/96**	**8:30 – 5:00**	**$1000-$1500***	**3996**
	Prerequisite(s): An understanding of and useful skills in a high-level programming language. A current ICN password is required. Topics Include: Introduction and Fundamentals; Basic Semantic Constructs - Getting; Base Level I/O With C; The Preprocess-Compilation Environment; Operators, Data Types, and Storage Classes; Control Flow Constructs; Conditional Constructs; Higher-Level Data Constructs in C; File I/O; UNIX Software Tools and POSIX System Calls.			
C Programming (Advanced)	**9/9-13/96**	**8:30 – 5:00**	**$1000-$1500***	**4777**
	Prerequisite(s): Useful skills and experience with the C Programming. A current ICN password is required. Topics Include: Data Structures, Algorithms, and OOP; An Advanced Clinic for C ; The ANSI C Recommendation X3.159; C and ANSI C War Stories; The Data Structure and the Assessment of Algorithms; Arrays; Structures; Unions; Stacks; Queues; Linked Lists; Recursive Functions; Binary Trees; Hashing; File Organizations Using the C Runtime Library; Standard Interprocess Communication Mechanisms; and An Introduction and Overview of AT&T's C++ 3.0.			
C++ for Experienced Programmers	**Available on Request (5 days)**		**$1000-$1500***	**9050**
	Prerequisite(s): Excellent C Language programming skills. Topics Include: Major Differences and Additions to ANSI C; Building C++ Classes; Introduction to Text I/O with C++; Function Overloading; Single Inheritance; Virtual Functions; Multiple Inheritance; Operator Overloading; Creating, Initializing and Assigning Objects; Passing and Returning Objects; Templates, Parameterized Functions and Classes; C++Stream I/O with the File System; and C++ Course Summary.			
C++ Visual Windows Programming	**7/22-26/96**	**8:30 – 5:00**	**$1600-$2000***	
	Prerequisite(s): C programming experience. Topics Include: Introduction to Visual C++; A Working Introduction to Windows Programming; Concepts of Object-Oriented Programming; Classes; Memory Management; Scope and Access Control; Functions in C++; References and Argument Passing; Operators; Class Design, Single Inheritance; Polymorphism and Virtual Functions; Microsoft Foundation Class Library; Windows Event Handling; The Mouse; The Keyboard; Graphics Device Interface; Dialog Boxes; Windows Memory Management; Menus and Accelerators; Document/View Architecture; MDI and Multiple Views; and Visual C++ and NT.			

Course Title	Date	Time	Cost	Course Number
Common Object Request Broker Architecture (CORBA) Seminar	**7/1-5/96**	**8:30 – 5:00 (4 days)**	**$1200-$1500***	**11563**
	Prerequisite(s): Familiarity with client/server environment; distributed, integrated applications and object oriented technology tools. Topics include: CORBA - A strategic overview; The future of CORBA; Technical introduction to CORBA; ORB interoperability; Universal Networked Objects (UNOs); Internet Interoperability Protocol (IIOP); The Dynamic Skeleton Interface (DSI); and Bridges.			
Perl Programming	**Available on Request (1-3 days)**		**$400-$600* per day**	**8095/8093**
	Topics Include: Describes the programming language that occupies the niche between shell and C Programming; syntax and semantics; data types; operators, control flow, regular expressions, and I/O facilities; the Perl debugger.			
Perl Programming for the WWW	**Available on Request (2-3 days)**		**$400-$600* per day**	
	Prerequisite(s): Programming skills with a light background in Perl and HTML. Topics Include: On-line Resources; Server Configuration; Permissions; Setuid Issues; Tainting; Safe Perl; Data Security; OO Programming; Web Modules; CGI Programs; CGI.pm; What Went Wrong?; CGI Template; Using Forms; Form Template; Input Widgets; Submit Widgets; Reset Widgets; Sample Form; Password Fields; Textareas; Hidden Fields; Checkboxes; Radio Boxes; Popup Menus; Lisboxes; Image Maps; Random Links; libwww Modules; Sending Mail; Shopping Carts; Database Access; and Advanced Topics.			
SGI System Administration (Beginning)	**7/8-12/96**	**8:30-5:00**	**$260**	**7993**
	Prerequisite(s): Familiarity with using Silicon Graphics IRIS workstations and system administration procedures on other open system platforms. Topics Include: The Role of the System Administrator; Set Up and Configuration of an IRIS Workstation or Server; Supporting a Group of Silicon Graphics Users; System Security Maintenance; Backups and Recoveries; Configuration of Disk Drives; System Installation and Application Software; Attaching Terminals and Printers; Modifying the system Start Up and Shut Down Sequences; Automating Administrative Procedures; and Performing Basic System Troubleshooting.			
SGI Network Administration	**8/26-30/96**	**8:30-5:00**	**$1700-$2200***	**11690**
	Prerequisite(s): Completion of Silicon Graphics System Administration (Beginning) course or equivalent knowledge and experience. Topics Include: Networking Fundamentals; Network Configuration; Network Troubleshooting; Resource Management with Network; Information Services; Domain Management with Domain Name System; Electronic Mail with Sendmail; Remote File Sharing with Network File System & Automounter; Network Performance Monitoring; and Network Security.			
SGI System Administration (Advanced)	**9/9-13/96**	**8:30 – 5:00**	**$1700-$2200***	**11689**
	Prerequisite(s): Completion of Silicon Graphics System Administration (Beginning) course or equivalent knowledge and experience. Topics Include: System Error Monitoring; Kernel Reconfiguration and Debugging; System Monitoring Tools; Process Management; MultiProcessor CPU Management; Memory Management and Tuning; Swap Management and Tuning; Disk Management and Tuning; XPS Filesystem Management; and System Security Concepts.			

Course Title	Date	Time	Cost	Course Number
Solaris 2.X Network Administration	**Available on Request (5 days)**		**$1300-$1700***	**8107**
	Note: This course was previously called Solaris 2.X System Administration (Advanced). Prerequisite(s): Completion of Solaris 2.X System Administration (Beginning) class or equivalent knowledge and experience. Topics Include: Network Configuration; Remote Installation Procedures; Advanced Security Techniques; Troubleshooting Techniques; Customizing Sendmail; network Application Tools; and Name Service Configuration.			
TCP/IP Internet-working on Windows NT	**Available on Request (4 days)**		**$1200-$1700***	
	Prerequisite(s): Completion of Windows NT Workstation and Server class or equivalent knowledge and abilities.Topics Include: Station-to-Station Communications; Connecting the Network; The TCP/IP Protocol Suite; IP Addresses; Subnets; TCP Utilities; System Configuration; SNMP; System Performance; IP Address Resolution; NetBIOS Name Resolution; Host Name Resolution; Implementing WINS; WINS Installation, Configuration, and Management; DHCP in Operation; Implementing DHCP; IP Routing Primer; Windows NT Routing; Heterogeneous Environments; TCP/IP Printing Support; and Installing the FTP Server.			
UNIX (Beginning)	**6/24-28/96**	**8:30 – 12:00**	**$738**	**5267**
	Prerequisite(s): Familiarity with a UNIX workstation. Topics Include: Overview of the Workstation Environment; Getting Started; The UNIX File System; Manipulating Files; Customizing Your Environment; The C-Shell; Editing and Writing with vi; Using the Network; Discussing NFS and NIS; Using Basic System Status Commands; Startup and Shutdown Procedures; Using tar.			
Volume Manager (VxVM) for System Administrators	**Available on Request (3 days)**		**$900-$1200***	
	Prerequisite(s): Knowledge of Solaris system administration is required.. Topics Include: Overview of Logical Volume Management; Introduction to VxVM Levels of Abstraction; Software Installation; Introduction to Administrative Interfaces; Software Initialization; Disk Configuration Management; Volume Configuration; Software Components; Administrative Command Utilities; Advanced Operations Through Low-Level Configuration Utilities; On-Line Reconfiguration Functionality; Analysis and Tuning; Conventions and Recommendations; and Failure Recovery and Other Issues.			
Windows NT Workstation and Server	**Available on Request (5 days)**		**$1300-$1800***	
	Prerequisite(s): This course is valuable for personnel who are evaluating or migrating to Windows NT. It benefits system and network administrators, other support personnel, programmers, and users from Windows, Unix, OS/2, or VMS backgrounds. Topics Include: Introduction to Windows NT; System Overview and Security; Network Configuration Options; Installation; Server Choices; User Administration and Security; Files and Printers; Built-in Network Support; Configuration Options; Using Setup; Data and Disk Management; The Registry; Troubleshooting; and Optimization and Performance.			
World Wide Web Development (Advanced)	**7/30/96 – 8/2/96**	**8:30 – 5:00**	**$1200-$1600***	**11526**
	Prerequisite(s): Prior knowledge of basic HTML, WWW servers, and browsers. Topics Include: Introduction; Advanced HTML; Netscape Advanced Features; Perl Programming; Common Gateway Interface (CGI); Quality Assurance Testing; Image Maps; Filers and Data Conversion Programs; Security; Graphical Tools; Internet Resources; Registration on the WWW; Statistics; Database Integration; Searching; Graphics; and Extended Data Types.			

Research Library Training

The LANL Research Library provides training for using its specialized databases. Training sessions begin and end at times indicated below. Classes are free but you must pre-register by calling the Research Desk at 7-5809 or sending e-mail to library@lanl.gov. Special classes and orientations can also be arranged.

Date	Time	Subject Matter
7-2-96	1:00-1:30 p.m.	Business Sources on the WWW
7-9-96	1:00-1:30 p.m.	SciSearch at LANL—At your desktop!
7-10-96	1:00-1:30 p.m.	Finding Addresses and Phone Numbers on the WWW
7-11-96	1:00-1:30 p.m.	1996 Chemical Abstracts on CD-ROM
7-11-96	2:00 - 4:00 p.m.	Information Sources on the Internet via WWW
7-16-96	1:00-1:30 p.m.	NTIS (US Govt Sponsored Research)—At your desktop
7-17-96	11:00 -11:30 a.m.	MELVYL (U of CA specialized databases)
7-17-96	1:00-1:30 p.m.	Finding Addresses and Phone Numbers on the WWW
7-18-96	1:00-1:30 p.m.	Science Sources on the WWW
7-23-96	1:00-1:30 p.m.	Federal Regulations
7-25-96	2:00 - 4:00 p.m.	Information Sources on the Internet via WWW
7-30-96	1:00 - 1:30 p.m.	1996 Chemical Abstracts on CD-ROM

Lab-Wide Systems Training

The Customer Service Group (CIC-6) offers training for users of Laboratory information systems. The CIC-6 courses offer training for a variety of personnel including property administrators, group secretaries, training coordinators, budget analysts, group leaders, or anyone needing to access training records, property records, costs, employee information, travel, chemical inventories, etc. Refer to the table below and on the following pages for specific information about courses currently offered.

Course Registration

You must have a valid ICN password before taking any of the courses shown in the table. To register for a course, call the CIC-6 Training, Development, and Coordination section at 667-9559 or access our Web page. From the LANL home page, look under "Services/Computing at LANL/Training" or enter the URL:

http://www.lanl.gov:8010/computer-information/cic6/teampage.html

Course Title	Date	Time	Cost	Course Number
Employee Development System - Basic Training (EDS I):	**7/3/96**	**8:30 – 12:00**	**$260**	**Course #5289**
	The course provides hands-on instruction to request course enrollment, use the on-line course catalog, retrieve training transcripts, and assign EDS authorities. The student will learn to create courses, add students to the courses, and generate several training reports.			
Employee Development System - Training Plans (EDS II):	**7/24/96**	**8:30 – 12:00**	**$260**	**Course #7155**
	Participants receive hands-on instruction to create and maintain training plans, assign assignment codes, and generate training plan reports. Attendees must have prior training in the Employee Development System (course #5289).			
Eudora Electronic Mail	**7/15/96**	**1:30 – 3:30**	**$130**	**Course #9762**
	This class is a hands-on class that teaches the participant how to use Eudora software to create, send, receive, and edit electronic mail messages. In addition to these procedures, the participant will learn what related settings mean and how to configure the system to meet his or her individual needs.			
Data Warehouse Basics	**7/23/96**	**8:30 – 10:30**	**$130**	**Course #11961**
	Students will receive hands-on training to generate standard reports and make quick queries from information in the data warehouse, a real-time collection of data tables from Laboratory financial, time-reporting, and personnel systems.			
Data Warehouse/ Financial Reporting	**7/2/96**	**8:30 – 12:00**	**$260**	**Course #11960**
	Students will receive hands-on training to generate standard financial reports and make on-line queries from information in the "data warehouse," a collection of data from Laboratory budgeting, accounting, and time-keeping systems.			

Course Title	Date	Time	Cost	Course Number
Financial Management Information System (FMIS):	**Scheduled on Request**		**$260**	**Course #8338**
	Participants receive hands-on instruction to "explode" and "transfer" through the costs, allocations, and outstanding commitments screens. In addition, participants will create/review reports, access the Information Manager Utility for printing reports, and learn how to assign authorities in the system.			
HTML Basics	**7/11/96**	**8:30 – 12:00**	**$260**	**Course #11605**
	Students will gain a basic understanding of HTML (Hypertext Markup Language), the language for the World Wide Web. Topics covered will be commands and standards, creating and editing documents, and authoring programs.			
HTML Tables	**7/26/96**	**8:30 – 12:00**	**$260**	**Course #11959**
	Students gain basic understanding of how to create various tables in HTML and new tags in HTML 3.0. Netscape-specific tags are also identified for clarity. Prerequisite: HTML Basics (Course #11605) or permission of the instructor.			
Introduction to the Internet: Beginning Netscape	**7/17/96**	**1:30 – 3:30**	**$130**	**Course #10961**
	Students gain basic understanding of the Internet and the World Wide Web and the use of Netscape as a browser to surf the Net. Topics covered are both Laboratory sites and open sites, along with practical uses of the Internet.			
Lotus Notes 4.0	**7/18/96**	**8:30 – 12:00**	**$260**	**Course #9917**
	This class provides hands-on instruction for Mac and PC users to use Lotus Notes software to create and send E-mail memos; fax documents; search databases; create filters, nicknames, banners, and doclinks; set defaults; and use multiple address books. In addition, participants learn how to use the memo, meetings, and discussion databases.			
On-Line Forms	**7/17/96**	**3:30 – 5:00**	**$130**	**Course #9756**
	Participants will learn to use Netscape software to access Lab-wide information and forms. Using Jetform Filler software, participants will access, complete, and print forms such as the "ICN Validation Request," "Visitor Request for Unclassified Visits to Security Areas," and "Request for Quotation."			
PCS Overview	**Scheduled on Request**		**no charge**	**Course #11924**
	Overview of Purchase Card System. Students will have taken BUS-5's credit card course. Call Ruby O'Rear at 665-4523 for course schedules.			

Course Title	Date	Time	Cost	Course Number
Property Accounting, Inventory, and Reporting System (Advanced)	**Scheduled on request**		**$260**	**Course #9918**
	This course will include a refresher of PAIRS, advanced techniques and tips, explanation of the notification system, and report capabilities. Swap Shop, Loan Out information, and support tables will be discussed. Participants should already have a basic understanding of and know how to use PAIRS.			
Purchase Card System	**7/22/96**	**8:30 – 9:30**	**no charge**	**Course #11924**
	Prerequisite: PCS Overview. Students will learn to reconcile monthly statement of account, submit reconciled statement of account for approval, print statement of account for audit records, and delegate reconciliation authority.			
Reporting with Infomaker	**7/9 – 10/96**	**8:30 – 5:00**	**$560**	**Course #11054**
	Hands-on training to query data and develop ad hoc, or non-standard, reports from the LANL data warehouse using Infomaker software.			
Time and Effort System (GUI)	**7/11/96**	**8:30 – 10:00**	**no charge**	**Course #11018**
	The student will learn how to enter attendance, amend attendance, approve attendance, and submit exception and approval reports. Time codes and associated policies will also be discussed. In addition, the student will learn how to use the Information Manager utility to view and print reports.			

CUT ALONG DASHED LINE

Los Alamos
National Laboratory

INTEGRATED COMPUTING NETWORK (ICN) VALIDATION REQUEST

To access ICN computing resources, please complete all parts of this form that apply to you, including "Special Requirements."

If you have **questions**: Call: (505) 665-1805
E-mail: validate@lani.gov

Mail your completed application to:
ICN Password Office (PWO)
Mail Stop: B271
Los Alamos National Laboratory
Los Alamos, NM 87545

All Laboratory computers, computing systems, and their associated communication systems are for official business only. By completing this request, users agree not to misuse the ICN. The Laboratory has the responsibility and authority to perodically audit user files.

Owner Information

Z-Number (if you have one)	PWO Use Only	Name (last, first, middle initial)
LANL Group	LANL Mail Stop	Citizenship (Foreign National see "Special Requirements-Foreign National")
Phone Number	Cost Center	Program Code

Check LANL affiliation:

- ☐ LANL employee
- ☐ Contractor _____ (specify contract company)
- ☐ Consultant, VSM, associate
- ☐ External user_____ (specify employer)
- ☐ Other (specify)_____

Send password / smartcard to:

☐ Mail Stop or ☐ Mail to address indicated below

Name / Organization
Address
City, State, Zip Code

Access

Check access method and needed partitions:

Access method: ☐ ICN Password ☐ Smartcard ☐ Both

☐ **Open** partition (e.g., email systems, open machines)

☐ **Administrative** partition (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMIS, PAIRS])
If you are not a Q-cleared LANL employee, see required steps in section "Special Requirements-Administrative Partition," unless you already have Administrative access with an ICN password.

☐ **Secure** partition (i.e., secure machines)
Indicate level(s) of data to be processed:
- ☐ Unclassified
- ☐ Secret

I certify this person does require secure access:

_____ _____
Manager Signature (Group Leader or above) Date

NOTE: A Q-clearance is required. All classified computing must be performed within the Secure environment.

PWO Use Only

New ☐ Change ☐	Clearance Status	Processed	Lv	Smartcard Serial #
Comments:				

Form 1646 (1/95) Supersedes previous versions (rev. 10/20).

Continue

Special Requirements

Administrative Partition
(U.S. Citizens Only) Lab-Wide Systems (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMIS, PAIRS])

☐ Under 18 years of age

If you need to access Administrative systems, your group leader must provide a memo accepting responsibility for your actions and justifying your need for access. This memo is to accompany all forms taken to the security briefing (see "Contractor or Non-Q-Cleared") section below. You may not access the Secure Partition.

☐ Contractor or Non-Cleared

Phone (505) 667-9444 to obtain Access Authorization packet.

Phone (505) 667-9153 to schedule a security briefing.

Bring all forms including this ICN Validation Request to the security briefing for approval.

Security Briefing Approval Signature	Date

☐ Foreign National

Attach a copy of Form 982 (REQUEST FOR UNCLASSIFIED VISIT OR ASSIGNMENT BY A FOREIGN NATIONAL) with all approval signatures. Be sure Box #11 of Form 982 is completed. If you are not a visitor/assignee under a LANL/DOE approved Visit / Assignment Request, attach written justification from your host Division Director describing your need to access the ICN.

Authorization (required)

Print Manager Name (Group Leader or above)	Manager Z-Number	Group
Manager Signature (Group Leader or above)	**Mail Stop**	**Date**

If you are NOT a LANL employee, obtain your LANL contact's signature in addition to the contact's manager's signature.
NOTE: LANL contacts are regular Laboratory employees. Contacts are responsible for obtaining annual re-authorizations, forwarding renewals, and notifying the ICN Password Office of changes in user or contact status.

Print LANL Contact Name	Contact Z-Number	Phone Number	Group
LANL Contact Signature		**Mail Stop**	**Date**

Reader Feedback

Feedback helps us to provide a document that responds to the changing needs of its readership. If you have comments or questions about this publication, please let us hear from you. We have reserved the back of this form for that purpose. We also accept articles for publication that are of interest to our readers. Contact the managing editor for more information. This form is also used for new subscriptions, deletions, or changes. Instructions are on the back. If you prefer to contact us by E-mail, send your comments and/or subscription request to finney@lanl.gov.

Do Not Staple
Fold on This Line First

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 88 LOS ALAMOS NM

POSTAGE WILL BE PAID BY THE ADDRESSEE

MAIL STOP B251
ATTN: MIKE FINNEY, MANAGING EDITOR
CUSTOMER SERVICE GROUP (CIC-6)
LOS ALAMOS NATIONAL LABORATORY
PO BOX 1663
LOS ALAMOS NM 87544-9916

cut along dashed line

Do Not Staple, Seal with Tape
Fold Here

Feedback

New Subscriptions, Deletions, and Changes

BITS is published by Los Alamos National Laboratory. If you would like to be added to or deleted from our mailing list, please check the appropriate line, complete the form below, and mail us the form.

_____ Add my name to the **BITS** mailing list.

_____ Delete my name from the **BITS** mailing list.

_____ Change my name/address as indicated below.

Name		Date
Address		Mail Stop
Group	Organization	
City	State	Zip
Phone	Number of copies	Employee Z#

INDEX

Produced by the Computing, Information, and Communications (CIC) Division
Managing Editor: Mike Finney (667-2241 or finney@lanl.gov)
Design: Gloria Sharp and Mike Finney
Printing: Media Group (CIC-17)

Los Alamos
NATIONAL LABORATORY

Los Alamos
NATIONAL LABORATORY

Los Alamos, New Mexico 87545

BITS is published monthly to highlight recent computing and communications activities within the Laboratory. We welcome your suggestions and contributions. BITS can be accessed electronically via Web browsers such as Mosaic and Netscape. Enter the following URL:

http://www.lanl.gov/Internal/divisions/cic/news.html

LALP-96-11 (6-96)